Requirements and Analysis Document for the Tetris project (RAD)

Version 0.2

Date 2011-03-30
Author Linus Karlsson, Andreas Karlberg, Magnus Junghard Huttu, Jonathan Kara

This version overrides all previous versions.

1 Introduction

This is an application of a game that maybe will look like a regular Tetris, but it has features that you've never seen before. You are in charge of a canon, which only purpose is to shoot the tetrominoes so that they fit together at the bottom. If the tetrominoes are creating a horizontal line without gaps, the line will disappear and you will get points. If you aren't fast enough the tetrominoes will stack on each other and when they reach the top of the field new tetrominoes aren't able to enter. If so happens, it's GAME OVER!

1.1 Purpose of application

The project aims to create a Tetris inspired game with a new touch. Instead of moving the tetrominoes, you move a cannon around the "Tetris-box" in which the tetrominoes are moving towards the bottom. To get points, one should fire laser balls with the cannon and shape them in a way so that they are fully covering a row.

1.2 General characteristics of application

The application will be a desktop, standalone (non-networked), single-player application with a graphical user interface for the Windows/Mac/Linux platforms.

The application will wait for the user to control the cannon. When the player is firing laser at the falling tetrominoes, the hit part of it will disappear. When they are hitting the bottom of the "Tetris-box", or if they get stacked on top each other, they will "freeze" and can not be removed. When the tetrominoes are "freezed" in a way that the cover a full row, the row will disappear and the player will receive points.

The application will end when the stacked blocks are reaching the top of the window, or possibly be canceled. If the game is cancelled, the player will be told to enter a nickname and his/her score will be saved.

1.3 Scope of application

The application does not include a computer-player. It is only possible to play the game alone. The application does save interrupted games or collects and saves the high scores.

1.4 Objectives and success criteria of the project

It should be possible to play a game where you can shoot tetrominoes away and also form them by shooting away small parts. There should be at least 2 sets of themes to change between.
The cannon should be easy to maneuver with the left and right arrows.
You should also be able to chose difficulties and watch high-scores on this graphical user interface.

1.5 Definitions, acronyms and abbreviations

- Tetris**,** an old invention of an russian mathematician who figured out a code for this game so that you could have some fun during boring times. Tetris is a game where you try to fit different shapes to eachother without leaving space, thus you need to fill a row with the shape-pieces and that will delete the row and give you an amount of points.
- Cannon, used to shoot down the blocks, and it will be moving on the outside of the tetris game panel. When your shot hits a tetris-block then a piece of it will vanish, you can continue to shoot it until it disappears or until you are satisfied with the shape.
- Tetrominoes, shootable blocks in the same shape as in the original game. When the shape reaches ground or if it touches an other shape, then it will lock-in and be unchangeable.
- Tetris box, where the tetris blocks will be falling. When the blocks land on the bottom of the box or on other blocks it stops falling.
- Freezed, the state when the blocks are unchangeable.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

The player should be able to:

1. Change settings
    a. Turn on/off sound
    b. Change theme
2. Look at old high-scores.
3. Start a new game
    a. Select level (beginner, intermediate, expert)
3. Control the cannon
    a. Move the around the “Tetris box” cannon with the keys.
    b. Shoot laser on the blocks to form them as you want.
5. Exit the application

2.2 Non-functional requirements

2.2.1 Usability

Usability is high priority. Normal users should be able to play the game within a very short period.

The application must be self-explanatory in such way that the user instantly feels superior, no matter if he/she is a experienced computer user or not. An English manual on how to play the game should be attached to the game.

2.2.2 Reliability

NA

2.2.3 Performance

Any action initiated by the player should immediately be updated on the screen, meaning that the response time is very low.

2.2.4 Supportability

The application must be implemented so that the GUI is easily modifiable to suit other platforms (Web, Mobile Apps, Pads, e.t.c.). There estimated time to adapt the GUI to at web based application should not exceed 1 man-month.

The implementation should prepare for the dividing of the application into a client/ serverarchitecture for net based games. It should be easy to partitioning the application into a client-server architecture. A time estimation for this should be included.

There should be automated test verifying all use cases. Code related to the GUI could be tested manually. GUI test should be recorded and included in the documentation

2.2.5 Implementation

To achieve platform independence the application will use the Java environment. All hosts must have the JRE installed and configured. The application needs to be installed on all hosts where it will run (possibly downloaded)

2.2.6 Packaging and installation

The application will be delivered as an zip-archive containing;

1. A file for the application code (a standard Java jar-file).
2. All needed resources, internationalization and localization les, icons, e.t.c.
3. Start programs (scripts) to start the game on the different platforms.
4. A README-file documenting installation and start of application.

2.2.7 Legal

There could be legal issues regarding rights to the Tetris game and trade mark. This is not covered here.

2.3 Application models

2.3.1 Use case model

see APPENDIX for UML diagram and textual descriptions

2.3.2 Use case priority

1. New Game
2. Falling Blocks
3. Move(move the cannon)
4. Shoot
5. Exit Game
6. On/off sound
7. Change theme

2.3.3 Domain model

see APPENDIX

2.3.4 User interface

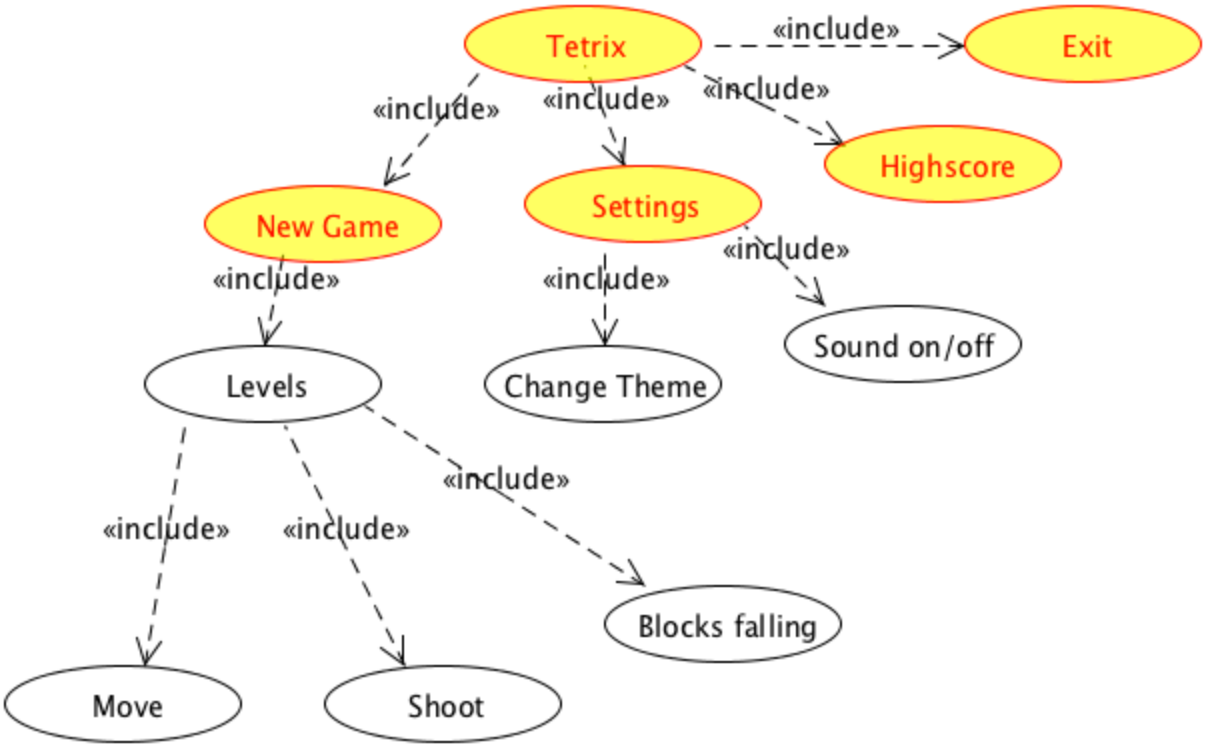
The application will be using a changeable GUI, where you can change things like the theme. The GUI will be offering a different amount of screen sizes. See APPENDIX for GUI.

2.4 References

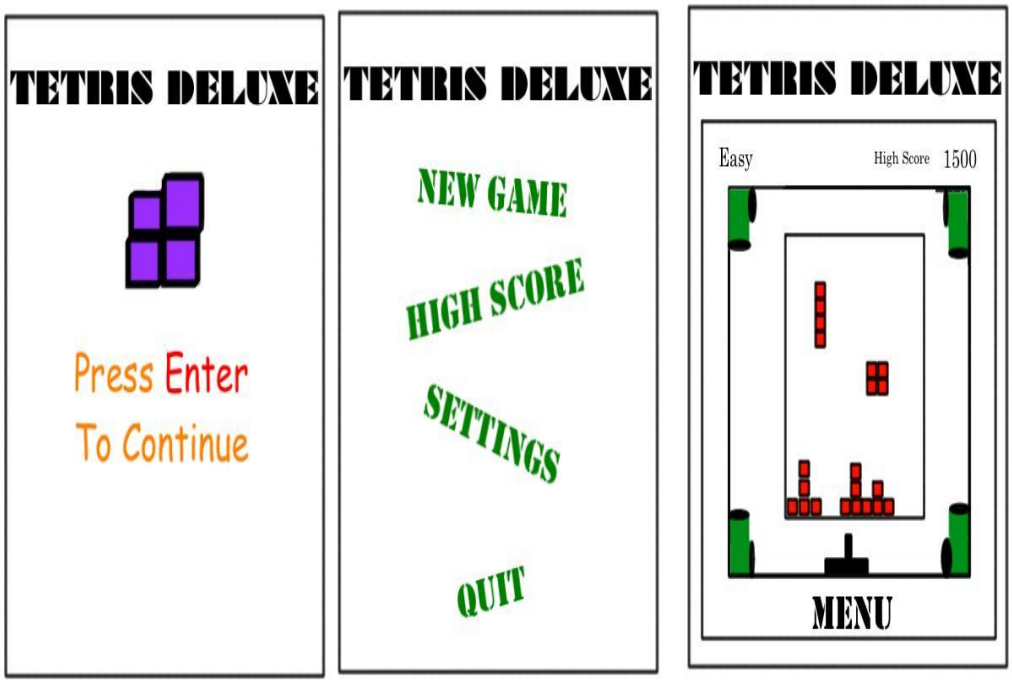
Tetris game: http://en.wikipedia.org/wiki/Tetris

Appendix

Use cases



GUI



Domain model

Use case: Move

Short description: How a user moves the cannon.
Priority: High
Extends or Includes: Shoot
Participating actors: Actual player (AP).

Normal flow of events

Actor	System
AP presses left- or right arrow	
	The cannon slides in given direction as long as the key is pressed.

Alternate flow: No alternate flow
Exceptional flow: No exception

Use case: On/off sound

Short description: How a player turns the sound on and off.
Priority: low.
Extends or includes: Extends GameSettings.
Participating actors: Sound, Player needs to be in pause settings menu or in main menu.
Normal flow of events

Actor	System
Actual Player clicks on Sound settings	
	The result is that two more options reveals themselves.
Actual Player clicks on Mute sound effects or Mute music.	
	The result is that either music or sound effects are muted.

Use case: Exit Game

Short description: How a quits the application
Priority: High
Extends or Includes: NA
Participating actors: Actual player (AP).

Normal flow of events

Actor	System
AP clicks the "Exit Game" button.	
	The application window shows a confirmation window. (Yes or No).

AP clicks YES or NO	
	The application either go back to the main menu, or shuts down.

Exceptional flow: No exception

Use case: New Game

Short description: How to a launch a new game.
Priority: High
Extends or Includes: NA
Participating actors: Actual player (AP).

Normal flow of events

Actor	**System**
AP clicks "New Game"-button	
	The program displays four different level choices (beginner, intermediate, expert and insane).
AP makes his/her choice	
	The game starts with the desired setting.

Use case: Blocks falling

Short description: These are some of the main objects in this game since they can make you lose the game, and win the game.
Priority: high.
Extends or includes: Extends StartNewGame.
Participating actors: Theme, level, freeze.
Normal flow of events

Actor	**System**
Actual Player clicks on New Game	
	The result is that a new game starts and blocks starts to fall
	The blocks fall and touches either another block or the bottom of the view

	The blocks stick and freeze

Use case: Shoot

Short description: How a user shoots with the cannon
Priority: Medium
Extends or Includes: NA
Participating actors: Actual player (AP)

Normal flow of events

Actor	**System**
AP presses space to shoot	
	The cannon fires a bullet in the direction of which the cannon is pointing.

Alternate flow: No alternate flow
Exceptional flow: No exception

Use case: Change theme

Short description: How a user changes the theme.
Priority: low.
Extends or includes: Extends GameSettings.
Participating actors: The colors of the game, The player needs to be in settings menu to be able to change the theme on the game.

Normal flow of events:

Actor	**System**
Actual Player clicks on Settings	
	The result is that more options reveals themselves.
Actual Player clicks Graphics settings	
	The result is that more Graphic settings reveals themselves.
Actual Player Changes the theme through a scrollbar.	

	The result is that the colors changes in the game.